 CMES, vol.114, no.1, pp.95-116, 2018

A Novel Interacting Multiple-Model Method and Its Application to Moisture Content Prediction of ASP Flooding

Shurong Li[1, *], Yulei Ge[2] and Renlin Zang[2]

Abstract: In this paper, an interacting multiple-model (IMM) method based on data-driven identification model is proposed for the prediction of nonlinear dynamic systems. Firstly, two basic models are selected as combination components due to their proved effectiveness. One is Gaussian process (GP) model, which can provide the predictive variance of the predicted output and only has several optimizing parameters. The other is regularized extreme learning machine (RELM) model, which can improve the over-fitting problem resulted by empirical risk minimization principle and enhances the overall generalization performance. Then both of the models are updated continually using meaningful new data selected by data selection methods. Furthermore, recursive methods are employed in the two models to reduce the computational burden caused by continuous renewal. Finally, the two models are combined in IMM algorithm to realize the hybrid prediction, which can avoid the error accumulation in the single-model prediction. In order to verify the performance, the proposed method is applied to the prediction of moisture content of alkali-surfactant-polymer (ASP) flooding. The simulation results show that the proposed model can match the process very well. And IMM algorithm can outperform its components and provide a nice improvement in accuracy and robustness.

Keywords: Interacting multiple model, regularized extreme learning machine, gaussian process, moisture content of ASP flooding.

1 Introduction

In practical applications, the industrial processes are always complex nonlinear dynamic systems, which are difficult and time-consuming to solve the mechanism models directly. With the development of computers, data-driven system identification can be viewed as an alternative way. Nonlinear system identification is to estimate models of nonlinear dynamic systems from observed input-output data, it is a subject with a long history of research interest and still remains active because of the need to accommodate more

[1] Automation School, Beijing University of Posts and Telecommunications, Beijing, China.

[2] College of Information and Control Engineering, China University of Petroleum (East China), Qingdao, China.

* Corresponding author: Shurong Li. Email: lishurong@bupt.edu.cn

CMES. doi:10.3970/cmes.2018.114.095 www.techscience.com/cmes

requirements and improve operating processes. Many traditional modeling methods are used in nonlinear system identification, such as nonlinear auto-regressive exogenous input (NARX) models [Hong, Tran and Yang (2010)], artificial neural network (ANN) models [Wang, Yan and Shi (2013)], support vector machine (SVM) models [Yan, Shao and Wang (2004)], Hammerstein models [Qi and Li (2008)], Wiener models [Qi and Li (2009)]. These methods have exposed several disadvantages in actual applications. Firstly, the prediction model often provide a scalar prediction only at any sampling point without any measure of the confidence in that prediction, because the models do not consider the uncertainty of model structure. A more suitable model should provide predictive error for each prediction, or even supply a complete predictive distribution. Secondly, overfitting problems often exist because much more attention is paid to reducing the errors at training points, especially in ANN models. GP models can be ideally suitable for solving the first problem. And RELM models can solve the second one. Moreover, model renewal caused by adding new meaningful samples can also help to alleviate the second problem.

GP models, also known as kriging in geostatistics, have developed rapidly in recent years because of the capability of providing the uncertainties of the predicted outputs and the relatively less number of optimizing parameters. O'Hagan et al. [O'Hagan and Kingman (1978)] first introduced the GP model approach to curve fitting. Later, Rasmussen [Rasmussen (1996)] compared GP models with other widely used models, leading to a rapidly growing attention to GP models. The paper by Rasmussen [Rasmussen (2006)] provided a long-needed, systematic and unified treatment of theoretical and practical aspects of GP models in machine learning. Ažman et al. [Ažman and Kocijan (2007)] used GP models for black-box modelling of Biosystems. A biomass concentration estimator for bath biotechnological processes by Bayesian GP regression is proposed di Sciascio et al. [di Sciascio and Amicarelli (2008)]. And Kocijan et al. [Kocijan and Likar (2008)] utilized GP models for gas-liquid separator modelling and simulation. Also, there are lots of variants about GP models, such as recursive GPR [Chan, Liu, Chen et al. (2013)], moving-window GPR [Ni, Tan, Ng et al. (2012)], sparse GP [Seeger, Williams, Lawrence et al. (2003); Csató and Opper (2002); Keerthi and Chu (2005)]. In general, GP models have been increasingly viewed as an alternative approach to other traditional modelling methods [Gregorčič and Lightbody (2008); Deisenroth, Zheng, Chen et al. (2009); Gregorčič and Lightbody (2009)].

Extreme learning machine (ELM) is a single-hidden layer feedforward neural network (SLFN) in fact, but unlike conventional one, ELM randomly chooses hidden nodes and analytically determines the output weights, leading to an extremely fast learning speed [Huang, Zhu and Siew (2006)]. Because of the good characteristics, ELM has been gradually used in nonlinear system identifications. A regression algorithm of quasi-linear model with ELM is proposed for nonlinear system identification [Li, Xie and Jin (2015); Li, Jia, Liu et al. (2014)] utilized OS-ELM for adaptive control of nonlinear discrete-time systems. A new method via ELM based Hammerstein model is proposed by Tang et al. [Tang, Li and Guan (2014)] for nonlinear system identification. Although with fast learning speed, ELM still can be considered as empirical risk minimization theme and tends to generate over-fitting model, so regularization method is introduced in many researches [e.g. Deng, Zheng and Chen (2009); Shao and Er (2016)]. Compared with

ELM, RELM greatly reduces the fluctuation caused by randomly generating input parameters and performs much better in generalization and stability.

For a nonlinear dynamic system, an invariable identification model is not suitable because the operating conditions often change in practical. It is necessary to continually update the identification model to track the process dynamics by incorporating new meaningful samples. Retraining the model is time-consuming once a new sample is added. So recursive methods can be regarded as nice ways to update models to reduce the computation burden. In recent years, many researches find that combing different methods is an effective way to improve prediction performances [e.g. Wang, Wang and Wei (2015); Yan, Shao and Wang (2004)]. Different models have different capabilities to capture data characteristics in linear and nonlinear domain. It seems reasonable to apply each models unique feature to capture different patterns in the data. Aiming at improving prediction performance, IMM is introduced to combine the advantages of GP models and RELM models in this study. IMM is recursive, modular and has fixed computational requirements per cycle, it has been demonstrated to be one of the most cost-effective and simple schemes for the estimation in hybrid systems [Zhang (2011)].

With oil development coming into tertiary oil recovery period, the growing attention has been paid to the technologies of enhancing oil recovery. As a new one, ASP flooding has a nice performance on enhancing oil recovery and obtained nice results in different oilfields. Some study show that ASP flooding pilot can form oil banks, greatly lower water cut, increase the oil production as well as the oil recovery. And the incremental oil recovery was about 20% over water flooding [Shutang and Qiang (2010)]. But the cost and risk of ASP flooding is higher compared with other technologies, such as water flooding, polymer flooding. So it is very important to optimize the ASP injection strategies in order to obtain the optimal economic benefit [Zerpa, Queipo, Pintos et al. (2005)]. And moisture content is a crucial index in the computation of economic benefit. However, ASP flooding process is a complex nonlinear distributed parameter system with strong spatiotemporal characteristic and uncertainty, and it is difficult and time-consuming to obtain moisture content by directly solving the mechanism model of ASP. Building a data-driven identification model can be viewed as an alternative way instead of solving the mechanism model.

The rest of the paper is organized as follows. In section 2, GP models and RELM models are introduced in brief. The meaningful new samples and recursive methods are presented in section 3. IMM algorithm is shown in section 4. Section 5 illustrates the simulation results of moisture content of ASP accompanied by discussion of the results. Finally, conclusions are drawn in section 6.

2 Model description

2.1 GP models

GP is a finite set of random variables with a joint Gaussian distribution. It provides a prediction of the output variables for a new input through Bayesian inference. Considering the output variable $\mathbf{y} = (y_1, \cdots, y_N)^T$, GP models the regression function having a Gaussian prior distribution with zero mean

$$\mathbf{y} = (y_1, \cdots, y_N)^{\mathrm{T}} \sim G(0, \mathbf{C}), \tag{1}$$

where $\mathbf{C}$ is the $N \times N$ covariance matrix whose elements are defined as $\mathbf{C}_{ij} = \mathrm{cov}(\mathbf{x}_i, \mathbf{x}_j)$. The following covariance function is one of the most commonly used in the literature

$$\begin{aligned} C(\mathbf{x}_i, \mathbf{x}_j) &= v_0 \exp[-\frac{1}{2}\sum_{d=1}^{D} w(x_i^d - x_j^d)^2] + a_0 \\ &+ a_1 \sum_{d=1}^{D} x_i^d x_j^d + \sigma_e^2 \delta_{ij}, \end{aligned} \tag{2}$$

where D is the dimension of input space of vector $\mathbf{x}_i = [x_i^1, \cdots, x_i^D]$. δ_{ij} is known as the Kronecker delta which is defined as

$$\delta_{ij} = \begin{cases} 1 & when\ i = j, \\ 0 & when\ i \neq j, \end{cases} \tag{3}$$

where $\boldsymbol{\theta} = [v_0, w, a_0, a_1, \sigma_e^2]$ is the vector of modeling parameters, also known as hyper-parameters. v_0 controls the overall scale of the local correlation and accounts for nonlinearity, which is similar to the form of radial basis function. w allows a different distance measure in each input dimension and σ_e^2 represents the estimate of the noise variance.

The hyper-parameters can be estimated by maximizing the log-likelihood using Bayesian inference

$$\begin{aligned} L(\theta) &= \log[\, p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{X})] = -\frac{1}{2}\log|\mathbf{C}| - \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{y} \\ &- \frac{N}{2}\log(2\pi), \end{aligned} \tag{4}$$

where $p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{X}) = G(0, \mathbf{C})$. The optimization problem can be solved based on the derivative of the log-likelihood corresponding to each hyper-parameter as

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = -\frac{1}{2} tr(\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \boldsymbol{\theta}}) + \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \boldsymbol{\theta}}\mathbf{C}^{-1}\mathbf{y}. \tag{5}$$

After the hyper-parameters are determined, the GP model can be obtained. When a new input $\mathbf{x}_*$ is given, the predictive distribution of y_* is

$$p(y_* \mid \mathbf{y}, \mathbf{X}, \mathbf{x}_*) = \frac{p(\mathbf{y}, y_*)}{p(\mathbf{y} \mid \mathbf{X})}. \tag{6}$$

And, its mean and variance are

$$\hat{y}_* = \mathbf{c}(\mathbf{x}_*)^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{y}, \tag{7}$$

$$\delta^2(\mathbf{x}_*) = c(\mathbf{x}_*) - \mathbf{c}(\mathbf{x}_*)^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{c}(\mathbf{x}_*), \tag{8}$$

where $\mathbf{c}(\mathbf{x}_*)$ is the covariance vector between new input and the training data, $c(\mathbf{x}_*)$ is the covariance of the new input. $\mathbf{c}(\mathbf{x}_*)^{\mathrm{T}}\mathbf{C}^{-1}$ can be viewed as smoothing term which weights the training outputs to predict y_* based on $\mathbf{x}_*$. Eq. (8) provides the confidence level of

the model prediction, where a higher variance means that the new data is further away from the training data and the prediction may be inaccurate.

2.2 RELM models

As a variant of ELM, RELM is similar to ELM in most aspects. A standard ELM can be represented as

$$y = \sum_{i=1}^{L} \beta_i g_i(\mathbf{x}) = \sum_{i=1}^{L} \beta_i G(\boldsymbol{\omega}_i, b_i, \mathbf{x}), \tag{9}$$

where L is the number of hidden nodes, y is output, and b_i is the threshold of the ith hidden node. β_i and $\boldsymbol{\omega}_i$ are the weights connecting the ith hidden node with the output nodes and input nodes respectively. For N distinct samples $(\mathbf{x}_j, y_j), \mathbf{x}_j = [x_j^1, \cdots, x_j^D], j = 1, 2, \cdots N$, Eq. (9) can be written as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T}, \tag{10}$$

where $\mathbf{H}$ and $\boldsymbol{\beta}$ are the hidden layer output matrix and output weight matrix respectively, $\mathbf{T}$ is target matrix, and

$$\mathbf{H} = \begin{bmatrix} G(\boldsymbol{\omega}_1, b_1, \mathbf{x}_1) & \cdots & G(\boldsymbol{\omega}_L, b_L, \mathbf{x}_1) \\ \vdots & \cdots & \vdots \\ G(\boldsymbol{\omega}_1, b_1, \mathbf{x}_N) & \cdots & G(\boldsymbol{\omega}_L, b_L, \mathbf{x}_N) \end{bmatrix}_{N \times L}, \tag{11}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_L \end{bmatrix}_{L \times 1}, \mathbf{T} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}_{N \times 1}. \tag{12}$$

Since the input weights and thresholds of the hidden nodes can be randomly selected, the output weight matrix $\boldsymbol{\beta}$ is the only parameter that needs to be calculated in the ELM. Aiming at minimizing $\| \mathbf{H}\boldsymbol{\beta} - \mathbf{T} \|^2$, $\boldsymbol{\beta}$ can be solved through the Least Square Estimate (LSE) method

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T}, \tag{13}$$

$$\mathbf{H}^{\dagger} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H}^{-1}\right)\mathbf{H}^{\mathrm{T}}, \tag{14}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix $\mathbf{H}$.

Ridge regression is one of the most common regularization methods in the RELM. According to the ridge regression theory, more stable and better generalization performance can be achieve by adding a positive value $1/C$ to the diagonal elements of $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ or $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ when calculating the output weights $\boldsymbol{\beta}$ [Toh (2008)]. Thus, Eq. (14) can be rewritten as

$$\mathbf{H}^{\dagger} = (\mathbf{H}^{\mathrm{T}}\mathbf{H}^{-1} + \frac{\mathbf{I}}{C})\mathbf{H}^{\mathrm{T}}, \tag{15}$$

which is the result of aiming at minimizing $\| \mathbf{H}\boldsymbol{\beta} - \mathbf{T} \|^2 + (1/C) \| \boldsymbol{\beta} \|^2$. Comparing with $\| \mathbf{H}\boldsymbol{\beta} - \mathbf{T} \|^2$, an extra penalty term $(1/C) \| \boldsymbol{\beta} \|^2$ is added to the target of RELM, which

makes RELM have better generalization ability.

The main procedure of RELM is described as follows:

1) Generate input weight $\boldsymbol{\omega}$ and threshold b of hidden nodes randomly.

2) Calculate the hidden layer output matrix $\mathbf{H}$ according to Eq. (11).

3) Calculate the output weight $\boldsymbol{\beta}$ through Eq. (13) and (15).

3 Model update schemes

In the practical processes, the varying operating conditions of the processes imply that the prediction based on invariant trained model is not suitable. When new samples become available, the models need to be retrained in order to provide a better prediction that reflects the real-time changes of the processes. However, it is a hard work to incorporate all new data for the model update, which will cause great computational load. In fact, it is not necessary to incorporate all new data because finite several new samples may provide most useful information for the model update. Therefore, data selection methods are developed to select meaningful new samples for updating the models. Moreover, recursive methods are adopted to reduce the computational burden.

3.1 Update scheme of GP models

GP models provide a prediction confidence level for the predicted output based on the new input. When the predictive variance $\delta(\mathbf{x}_*)$ is small, the prediction can be considered as accurate. And the prediction may not be accurate if the predictive variance $\delta(\mathbf{x}_*)$ is large, which also means the new data $\mathbf{x}_*$ is regarded as meaningful for the model update. So a variance limit $\delta_{\text{limit}}(\mathbf{x}_*)$ needs to be set to select the meaningful new data. If the predictive variance is larger than $\delta_{\text{limit}}(\mathbf{x}_*)$, the new data can be viewed as meaningful. If the predictive variance is smaller than $\delta_{\text{limit}}(\mathbf{x}_*)$, the new data is considered as useless.

In the process of adding new data, the recursive method [Chan, Liu and Chen (2013)] is utilized for the model update. Assuming that N data set is initially used to train the model, after adding additional data with a subscript $N+1$, the predictive mean and variance can be written as

$$\hat{y} = \mathbf{c}(\mathbf{x}_{N+1})^{\mathrm{T}} \mathbf{C}_{N+1}^{-1} \mathbf{y}_{N+1}, \tag{16}$$

$$\delta^2(\mathbf{x}_{N+1}) = c(\mathbf{x}_*) - \mathbf{c}(\mathbf{x}_{N+1})^{\mathrm{T}} \mathbf{C}_{N+1}^{-1} \mathbf{c}(\mathbf{x}_{N+1}), \tag{17}$$

where $\mathbf{C}_{N+1}$ is the updated covariance of the input data, the vectors $\mathbf{c}(\mathbf{x}_{N+1})$ and $\mathbf{y}_{N+1}$ are defined as

$$\mathbf{c}(\mathbf{x}_{N+1}) = [C(\mathbf{x}_*, \mathbf{x}_1)\, C(\mathbf{x}_*, \mathbf{x}_2) \cdots C(\mathbf{x}_*, \mathbf{x}_N)]^{\mathrm{T}}, \tag{18}$$

$$\mathbf{y}_{N+1} = [y_1 \cdots y_{N+1}]^{\mathrm{T}}. \tag{19}$$

Here, $\mathbf{C}_{N+1}^{-1}$ is obtained through the available $\mathbf{C}_N^{-1}$ instead of using all the available $N+1$ data to recalculate the covariance function. According to Sherman-Morrison-Woodbury [Golub and Van Loan (2012)], the covariance matrix can be presented as

$$\mathbf{C}_{N+1}^{-1}=\begin{bmatrix}\mathbf{C}_N^{-1} & \mathbf{0}\\ \mathbf{0} & 0\end{bmatrix}+\mathbf{r}_{N+1}\mathbf{r}_{N+1}^{\mathrm{T}}z_{N+1}, \tag{20}$$

where the new covariance matrix $\mathbf{C}_{N+1}^{-1}$ is equal to the old covariance matrix $\mathbf{C}_N^{-1}$ plus a new updating term about the new data. In Eq. (20),

$$\mathbf{r}_{N+1}=[\mathbf{p}_{N+1}\mathbf{C}_N^{-1}-1]^{\mathrm{T}}, \tag{21}$$

$$z_{N+1}=\frac{1}{p_{N+1}-\mathbf{p}_{N+1}^{T}\mathbf{C}_N^{-1}\mathbf{p}_{N+1}}, \tag{22}$$

$$\mathbf{p}_{N+1}=[C(\mathbf{x}_1,\mathbf{x}_{N+1})\cdots C(\mathbf{x}_N,\mathbf{x}_{N+1})]^{\mathrm{T}}, \tag{23}$$

$$p_{N+1}=C(\mathbf{x}_{N+1},\mathbf{x}_{N+1}). \tag{24}$$

When the new data becomes available, the new covariance matrix $\mathbf{C}_{N+1}^{-1}$ can be updated recursively. The mean and variance of the predicted output can be obtained through Eq. (16) and (17).

3.2 Update scheme of RELM models

Unlike GP models, RELM cannot provide the confidence level for the prediction. Thus, the other simple data selection method is introduced. Considering the predictive error

$$e=\hat{y}-y, \tag{25}$$

It can be viewed as a measure of data selection. An error limit e_{limit} is set to determine if new data would be incorporated in the model update. If the predictive error is larger than e_{limit}, the new data can be regarded as meaningful new data which would benefit the performance of a model. On the contrary, the new data would be useless for the model update.

Assume that N data set is initially used to train the model to satisfy $N \geq L$. According to the research by Shao et al. [Shao and Er (2016)], a recursive method can be adopted to update the model. We define K_N^{-1} as follows

$$\mathbf{K}_N^{-1}=[\mathbf{H}_N^{\mathrm{T}}\mathbf{H}_N+\frac{\mathbf{I}}{C}]^{-1}, \tag{26}$$

where $\mathbf{H}_N$ is initial output matrix of hidden layer.

When the meaningful new data (one or more) come in, the new output matrix of hidden layer becomes

$$\mathbf{H}_{N+1}=\begin{bmatrix}\mathbf{H}_N\\ \delta\mathbf{H}_{N+1}\end{bmatrix}, \tag{27}$$

where $\delta\mathbf{H}_{N+1}$ can be obtained from new data. At the meanwhile, the inverse K_{N+1}^{-1} can be calculated as follows

$$\begin{aligned}\mathbf{K}_{N+1}^{-1} &= [\mathbf{H}_{N+1}^{\mathrm{T}}\mathbf{H}_{N+1} + \frac{\mathbf{I}}{C}]^{-1} \\ &= [\mathbf{H}_{N}^{\mathrm{T}}\mathbf{H}_{N} + \frac{\mathbf{I}}{C} + \delta\mathbf{H}_{N+1}^{\mathrm{T}}\delta\mathbf{H}_{N+1}]^{-1} \\ &= [\mathbf{K}_{N} + \delta\mathbf{H}_{N+1}^{\mathrm{T}}\delta\mathbf{H}_{N+1}]^{-1}\end{aligned} \tag{28}$$

According to the Woodbury matrix identity [Golub and Van Loan (2012)], Eq. (27) can be decomposed further.

$$\begin{aligned}\mathbf{K}_{N+1}^{-1} &= [\mathbf{K}_{N} + \delta\mathbf{H}_{N+1}^{\mathrm{T}}\mathbf{I}\delta\mathbf{H}_{N+1}]^{-1} \\ &= \mathbf{K}_{N}^{-1} - \mathbf{K}_{N}^{-1}\delta\mathbf{H}_{N+1}^{\mathrm{T}}[\mathbf{I} + \delta\mathbf{H}_{N+1} \\ &\quad \mathbf{K}_{N}^{-1}\delta\mathbf{H}_{N+1}^{\mathrm{T}}]\delta\mathbf{H}_{N+1}\mathbf{K}_{N}^{-1}\end{aligned} \tag{29}$$

The output weight $\boldsymbol{\beta}_{N+1}$ can be calculated as follows

$$\boldsymbol{\beta}_{N+1} = \mathbf{K}_{N+1}^{-1}\mathbf{H}_{N+1}^{\mathrm{T}}\mathbf{T}_{N+1}, \tag{30}$$

where $\mathbf{T}_{N+1} = \begin{bmatrix} \mathbf{T}_{N} \\ \delta\mathbf{T}_{N+1} \end{bmatrix}$, $\mathbf{T}_{N}$ is initial target matrix, and $\delta\mathbf{T}_{N+1}$ is new target data.

With meaningful new data coming in continually, the matrix $\mathbf{K}_{N+1}^{-1}$ can be updated recursively. The output weight $\boldsymbol{\beta}_{N+1}$ can be updated according to Eq. (29).

In the model update schemes, several problems need to be noticed. Firstly, the predefined variance limit $\delta_{\mathrm{limit}}(\mathbf{x}_{*})$ and error limit e_{limit} need to be set according to actual conditions. The lower predefined variance limit means more meaningful new samples need to be added in the model update, and the computational burden will be larger. Secondly, there are parameters which are not updated whether in the GP model (hyper-parameters) or in the RELM model (input weight and hidden threshold). With the process going on, the predictive accuracy may not meet the requirement. So, it is necessary to update these invariant parameters (retrain) when it happens. Thirdly, the train data will continue to grow with the addition of new data, causing large computational burden. A reasonable method to remove redundant data needs to be considered. And this paper does not focus on these problems.

4 Hybrid prediction based on IMM algorithm

Single-model predictor has a given prediction accuracy, while different models have different ability to adapt the changes from systems, so hybrid prediction is considered to improve the prediction accuracy and robustness. The concrete form is shown as

$$y = \omega_1 y_1 + \omega_2 y_2, \tag{31-a}$$

$$y_1 = \mathbf{c}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{y}, \tag{31-b}$$

$$y_2 = \sum_{i=1}^{L} \beta_i G, \tag{31-c}$$

where ω_1, ω_2 are combination weights of different models, y_1, y_2 are outputs from GP models and RELM models respectively. In this section, IMM algorithm is used to

identify the combination weights of different models, and similar applications of the algorithm can be seen in Wang et al. [Wang, Qi, Yan et al. (2016)] and Zhang [Zhang (2011)]. The algorithm is proposed under the multiple model estimation, a model set contains limited models $M = \{m^j\}\ j = 1, 2, \cdots, r$, where model m^j describes the corresponding mode of different model output, r is the possible number of system modes. In multiple model estimation, the system mode sequence is assumed as a Markov chain, and the model transformation accords with Markov process.

Generally, a discrete nonlinear system can be defined as

$$\begin{cases} x_k = f(x_{k-1}, u_k) + \sigma_k, \\ y_k = h(x_{k-1}) + \mu_k, \end{cases} \tag{32}$$

where x_k is state vector, u_k is input vector, y_k is measure vector, σ_k is process noise with mean zero and covariance Q_k , μ_k is measure noise with mean zero and covariance R_k . If let x_k , y_k represent the predicted output and measure output respectively, then the prediction of moisture content of ASP can be viewed as a similar system defined as

$$\begin{cases} x_k = f(x_{k-1}, u_k) + \sigma_k, \\ y_k = x_k + \mu_k, \end{cases} \tag{33}$$

Considering this point, it is very cost-effective to make a hybrid prediction for moisture using IMM algorithm. The IMM algorithm in this paper can be divided into four steps:

1) Model-conditional re-initialization

Suppose that matching models are m^i and m^j at time $k-1$ and k , the mixing probability conditioned on y_{k-1} is

$$\omega_{k-1|k-1}^{i|j} = P[m_{k-1|k-1}^{i|j}, y_{k-1}] = \frac{1}{\bar{c}_j} \pi_{ij} \omega_j^{k-1}, \tag{34-a}$$

$$\bar{c}_j = \sum_{j=1}^{r} \pi_{ij} \omega_j^{k-1},\ i, j = 1, 2, \cdots, r, \tag{34-b}$$

where ω_j^{k-1} refers to the probability of jth model at time $k-1$, π_{ij} is the Markov transition probability, $\bar{c}_j$ is a normalization factor.

For $j = 1, 2, \cdots, r$, the mixed estimations of re-initialization state and its covariance are

$$\hat{x}_{k-1|k-1}^{oj} = E[x_{k-1} \mid m_k^j, y_{k-1}] = \sum_{i=1}^{r} \omega_{k-1|k-1}^{i|j} \hat{x}_{k-1|k-1}^{i}, \tag{35}$$

$$\begin{aligned} P_{k-1|k-1}^{oj} = \sum_{i=1}^{r} \omega_{k-1|k-1}^{i|j} [P_{k-1|k-1}^{i} + (\hat{x}_{k-1|k-1}^{i} - \hat{x}_{k-1|k-1}^{oj}) \\ \bullet (\hat{x}_{k-1|k-1}^{i} - \hat{x}_{k-1|k-1}^{oj})^{\mathrm{T}}], \end{aligned} \tag{36}$$

2) Model-conditional unscented Kalman filtering (UKF)

After the estimation of re-initialization state and covariance, UKF is utilized to update state estimate with new measure output y_k. In fact, y_k is not available at time $k-1$, here it is replaced by y_{k-1}.

(a) State sampling

$$x_{k-1}^j = \Big[\hat{x}_{k-1|k-1}^{oj}, \quad \hat{x}_{k-1|k-1}^{oj} + \sqrt{(n+\lambda)P_{k-1|k-1}^{oj}}, \quad \hat{x}_{k-1|k-1}^{oj} + \sqrt{(n+\lambda)P_{k-1|k-1}^{oj}} \Big], \tag{37}$$

where n is the dimensionality of state x, λ is the coefficient factor which can be learned more in references about UKF.

(b) Time update

$$\tilde{x}_{i,k-1|k-1}^j = f(x_{i,k-1}^j, u_{k-1}) \ \ i = 0,1,\cdots,2n, \tag{38}$$

$$\hat{x}_{k|k-1}^j = \sum_{i=0}^{2n} W_i^{(m)} \tilde{x}_{i,k-1|k-1}^j \ \ i = 0,1,\cdots,2n, \tag{39}$$

$$P_{k|k-1}^j = \sum_{i=0}^{2n} W_i^{(c)} (\tilde{x}_{i,k|k-1}^j - \hat{x}_{k|k-1}^j)(\tilde{x}_{i,k|k-1}^j - \hat{x}_{k|k-1}^j)^{\mathrm{T}} + Q_k, \tag{40}$$

$$W_0^{(m)} = \frac{\lambda}{n+\lambda} \ \ i = 0, \tag{41}$$

$$W_0^{(c)} = \frac{\lambda}{n+\lambda} + (1-\alpha^2+\beta) \ \ i = 0, \tag{42}$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(n+\lambda)} \ , i = 1,2,\cdots,2n. \tag{43}$$

(c) Measure update

$$S_k^j = P_{k|k-1}^j + R_k, \tag{44}$$

$$K_k^j = P_{k|k-1}^j (S_k^j)^{-1}, \tag{45}$$

$$\varepsilon_k^j = y_k - \hat{x}_{k|k-1}^j, \tag{46}$$

$$\hat{x}_{k|k}^j = \hat{x}_{k|k-1}^j + K_k^j \varepsilon_k^j, \tag{47}$$

$$P_{k|k}^j = P_{k|k\text{-}1}^j - K_k^j P_{k|k\text{-}1}^j, \tag{48}$$

where K_k^j is the UKF gain of *j*th model at time k, ε_k^j is the UKF residual of *j*th model at time k.

3) Model probability update

The model probability can be calculated by

$$\omega_k^j = P[m_k^j \mid y_k] = \frac{1}{c} \Lambda_k^j \bar{c}_j, \tag{49}$$

where $c = \sum_{j=1}^{r} \Lambda_k^j \bar{c}_j$. Λ_k^j is the likelihood function matching model m_j at time k, it can be obtained by

$$\begin{aligned} \Lambda_k^j &= P[y_k \mid m_k^j, y_{k-1}] \\ &= (2\pi \mid S_k^j \mid)^{-\frac{1}{2}} \exp(-\frac{1}{2}(\varepsilon_k^j)^{\mathrm{T}} (S_k^j)^{-1} \varepsilon_k^j). \end{aligned} \tag{50}$$

4) Predicting combination

The combination result and its covariance at time $k+1$ are

$$y_k^{\mathrm{IMM}} = \hat{x}_{k|k} = \sum_{j=1}^{r} \omega_k^j \hat{x}_{k|k}^j, \tag{51}$$

$$P_k^{IMM} = P_{k|k} = \sum_{j=1}^{r} \omega_k^j P_{k|k}^j + \sum_{j=1}^{r} \omega_k^j [(\hat{x}_{k|k}^j - x_{k|k})(\hat{x}_{k|k}^j - x_{k|k})^{\mathrm{T}}]. \tag{52}$$

The final prediction y_k^{IMM} is probability-weighted sum of prediction from all models which are obtained in the data-driven model identification.

As it is shown in Fig. 1, the overall structure of the proposed predictor can be divided into three parts. Firstly, initial data set is used to train GP model and RELM model respectively. Secondly, the predicted values from initial GP model and RELM model are compared with predefined accuracy. If the accuracy meets the requirement, the predicted values can go into the IMM predictor directly. Otherwise, meaningful new data selected by data selection methods are introduced to update the GP model and RELM model recursively. Then new predicted values from updated models go into the IMM predictor. At the last part, the predicted values from different models are weighted according to the probability to obtain the final predicted values in IMM predictor.

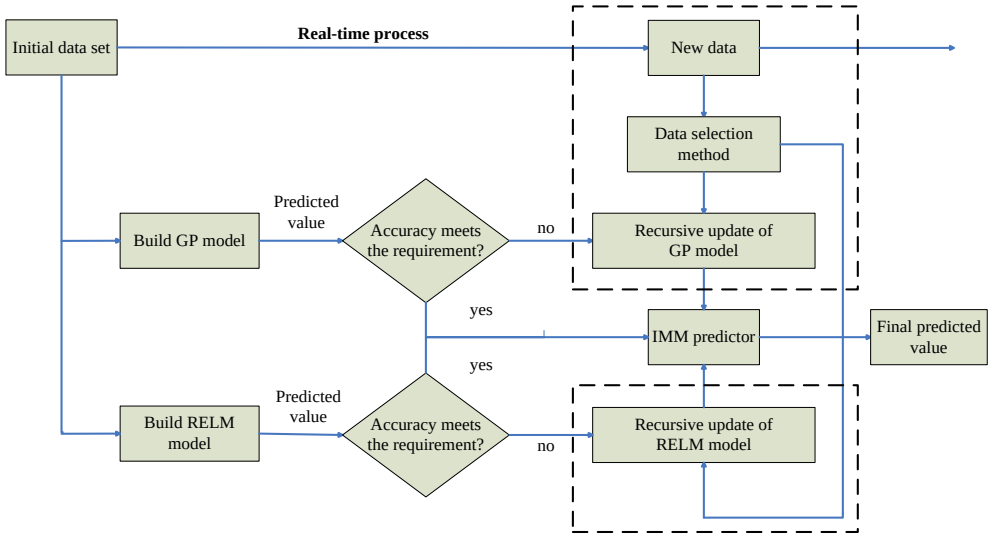

Figure 1: The overall structure of the proposed prediction method

5 Application to moisture content of ASP flooding

In this section, the characteristic of the proposed method and its application will be discussed. Firstly, the ASP flooding process is introduced. It is noted that we just list the one-dimensional model because three-dimensional model for ASP flooding that includes a series of divergence and cross terms, is too complex. The model for ASP flooding is

$$\frac{\partial S_w}{\partial t} = -\frac{Q}{A\phi}\frac{\partial f_w}{\partial x}, \tag{53-a}$$

$$\phi S_w \frac{\partial C_a}{\partial t} = -v_w \frac{\partial C_a}{\partial x} + \frac{\partial}{\partial x}\left(D_a \phi \frac{\partial C_a}{\partial x} \right) - \rho_r \frac{\partial \Gamma_a}{\partial t} - R_a, \tag{53-b}$$

$$\phi S_w \frac{\partial C_s}{\partial t} = -v_w \frac{\partial C_s}{\partial x} + \frac{\partial}{\partial x}\left(D_s \phi \frac{\partial C_s}{\partial x} \right) - \rho_r \frac{\partial \Gamma_s}{\partial t}, \tag{53-c}$$

$$\phi S_w \frac{\partial C_p}{\partial t} = -v_w \frac{\partial C_p}{\partial x} + \frac{\partial}{\partial x}\left(D_p \phi \frac{\partial C_p}{\partial x} \right) - \rho_r \frac{\partial \Gamma_p}{\partial t}, \tag{53-d}$$

where a denotes the alkali, s denotes the surfactant, p denotes the polymer, S_w is the water saturation, A is the core cross section area, f_w is the moisture content, v_w denotes the seepage speed of water phase, C_a, C_s, C_p and D_a, D_s, D_p denote the concentration and diffusion coefficient of alkali, surfactant and polymer respectively. ρ_r denotes the core density, $\Gamma_a, \Gamma_s, \Gamma_p$ are the adsorbing capacity of core for different displacing agents, and R_a is the alkali consumption.

The aim is to build the relationship between moisture content f_w and injection concentration C_a, C_s, C_p. Obviously, it is a nonlinear dynamic system which is very difficult and time-consuming to solve the mechanism model for ASP flooding directly. So the data-driven identification model is built in this paper. The three-slug injection is adopted to calculate the moisture content in this case. The injection concentration is

$C_a = u_a = (3.5, 2.4, 2.3)$, $C_s = u_s = (2.4, 1.2, 0.5)$,

$C_p = u_p = (2.3, 1.2, 0.7)$,

where $\mathbf{u} = [u_a, u_s, u_p]$ means the normalized injection concentration. In the identification model, the output is moisture content, represented as y, and the input vector is presented as $\mathbf{x} = [y(k-1), \mathbf{u}(k)]$. There are four injection wells with the same injection concentration, nine production wells, and the center of every four production wells is an injection well. The distribution of well position is shown in Fig. 2, where “S” means production well, “I” means injection well.

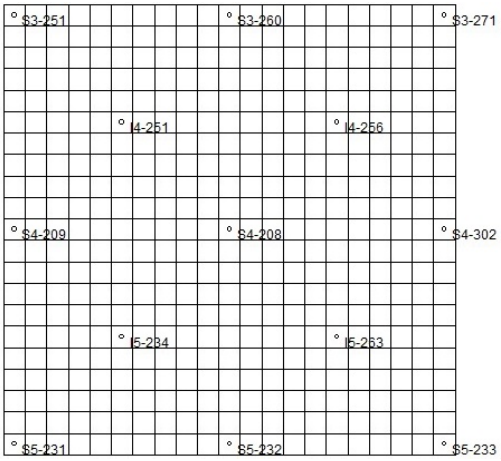

Figure 2: The distribution diagram of well positon

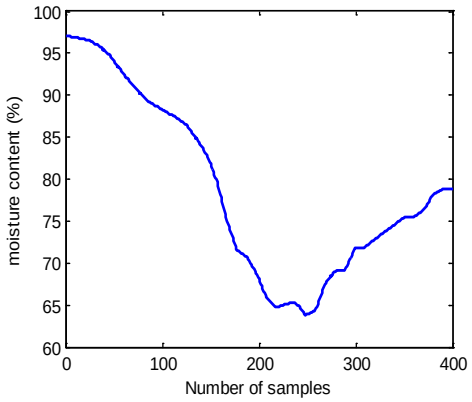

Figure 3: Moisture content data sets from production well S3-251

In this paper, the data from production well S3-251 is chose for model construction. In fact, data from other production wells can also be used for model construction. The moisture content can be calculated from reservoir simulation software CMG. About 1600 samples are obtained, one is retained every four samples, so 400 samples are left, where the first 200 is training data, the other 200 is testing data. Moreover, in order to model the real measure data more reasonable, the random white noise with appropriate size can be added to the final 400 samples. The final 400 samples are shown in Fig. 3 accompanied with corresponding injection concentration in Fig. 4.

The modeling performance is measured by the degree of agreement between the actual process and the model predicted output. In general, in order to show the efficiency of models, the comparison may be performed through visual inspection of the responses between the model and the process, then quantitatively evaluate the performance based on the values of selected performance indicators. The common performance indicators are root-mean-square error (RMSE) and relative root-mean-square error (RRMSE). If there are K elements, they are defined as

$$\text{RMSE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(y_k - \hat{y}_k)^2}\ , \tag{54}$$

$$\text{RRMSE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(\frac{y_k - \hat{y}_k}{y_k}\right)^2}\ , \tag{55}$$

RMSE is a good measure of accuracy for a particular variable, but not between variables, because it is scale-dependent. RRMSE gives relative ratios without units, which are usually expressed as percentages. In addition, the absolute value of the relative error (ARE) is employed to evaluate the capability of model to track the trend of an evolving process.

$$\text{ARE} = \left|\frac{y_k - \hat{y}_k}{y_k}\right| \tag{56}$$

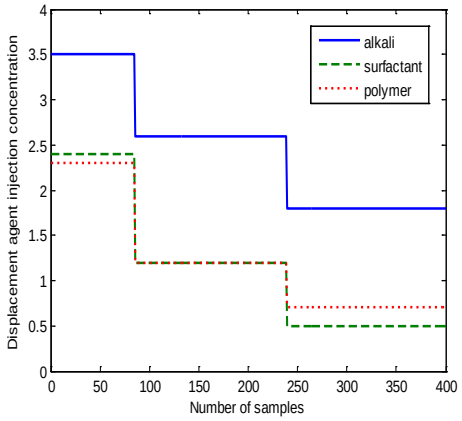

Figure 4: Displacement agent injection concentration for sample data

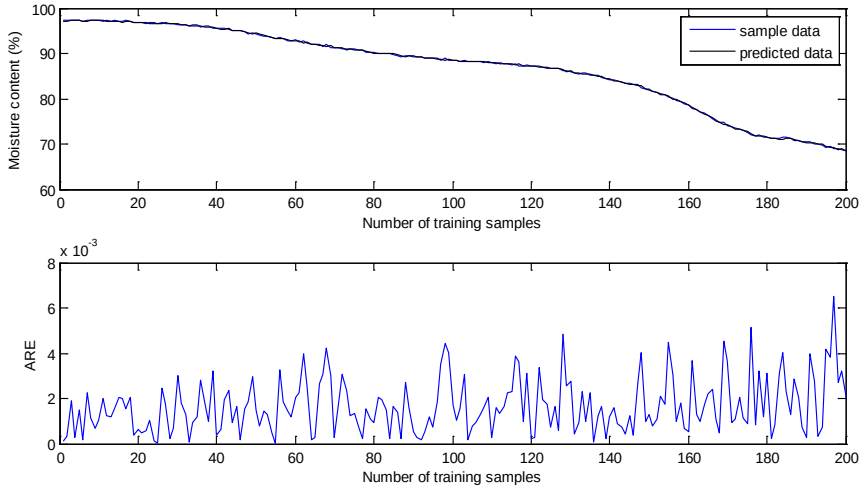

Figure 5: Predicted results of GP model on training samples

The first 200 samples are utilized to train the GP model. The training results are shown in Fig. 5. Most of the ARE are below 0.004, the RMSE and RRMSE are 0.1726 and 0.21% respectively. It is no doubt that the GP model matches the training samples very well. In order to verify the prediction performance, the obtained GP model is tested on the other 200 samples. Fig. 6 shows the predicted results of GP model on testing samples without update. It is clear that the predicted data curve gradually deviates from sample data curve at about the 40*th* testing sample. The predictive variance curve also appears a big rise at the same point, which means the prediction may be not accurate. In fact, it can be found from Fig. 4 that the control vector shows a sudden fall at about the 40*th* testing sample (the 240*th* sample). However, the training samples do not include the information of the sudden change of control vector. It can be seen that the changing operating conditions affect the prediction accuracy seriously. According to the proposed method, the predictive variance is above the variance limit, the corresponding data can be viewed as meaningful new data, which will be used to update GP model recursively. Fig. 7 demonstrates the predicted results after the first update with the addition of new data (the 240*th* sample). Compared with Fig. 6, the predictive variance becomes smaller, and the prediction performance improves a lot. If the predictive accuracy meets the accuracy requirements (predefined according to the practical situation), the update process will be terminated, otherwise other meaningful new data will be selected like the first one through the predictive variance until the accuracy requirements are met. Because the middle update processes are similar to the first one, so we do not enumerate here. The predicted results of GP on testing samples after the second update are demonstrated in Fig. 8. It is obvious that the predictive variances are all very small, denoting a high confidence level of the prediction. In addition, the RMSE and RRMSE of update process are listed in Tab. 1. It can be seen that the prediction performance has a big improvement after each update.

Table 1: RMSE and RRMSE of GP model for prediction

GP	Without update	After the first update	After the second update
RMSE	4.5783	1.1937	0.4126
RRMSE (%)	6.26	1.57	0.41

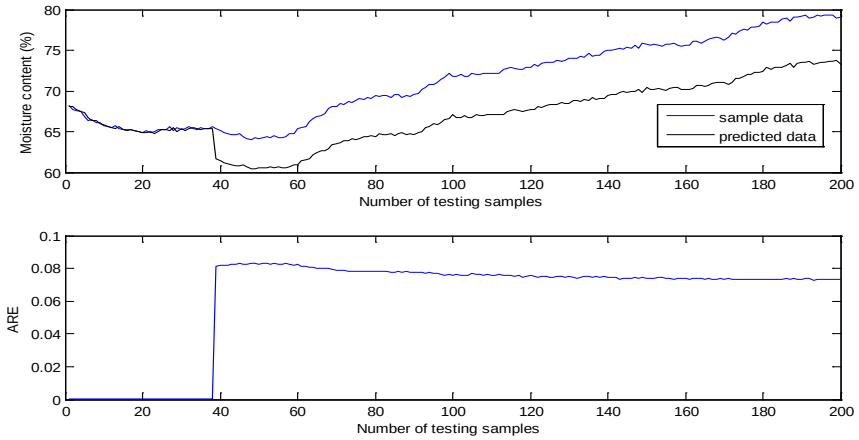

Figure 6: Predicted results of GP model on testing samples without update

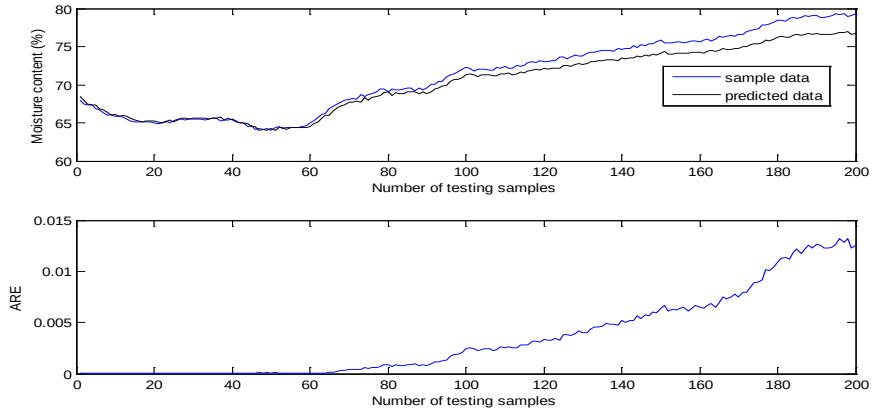

Figure 7: Predicted results of GP model on testing samples after the first update

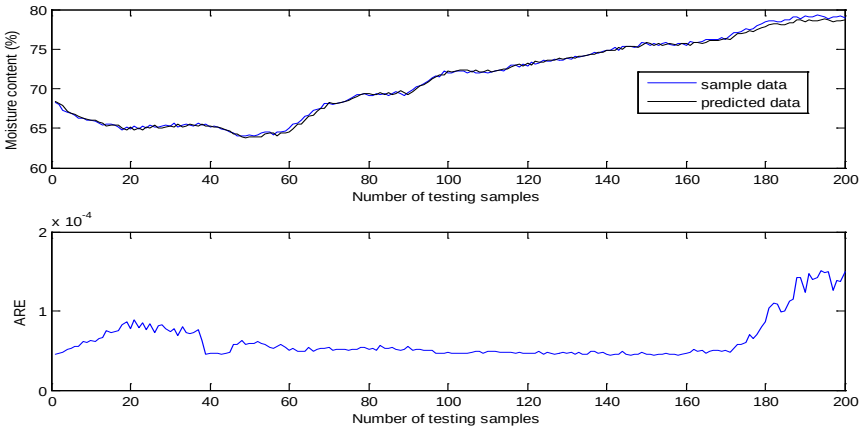

Figure 8: Predicted results of GP model on testing samples after the second update

The training results of RELM model is shown in Fig. 9. The model also shows a good match with the training samples. The RMSE and RRMSE are 0.2137 and 0.25% respectively. As shown in Fig. 10, a sudden change also occurs at about the 40*th* testing sample because of the same reason mentioned above. The difference is that predictive errors (ARE) become the measure of data selection. The predicted results of the first update and the second update are presented in Fig. 11 and 12, the other middle update processes are also omitted. The RMSE and RRMSE are summarized in Tab. 2. A similar conclusion can be drawn that the update improves the predictive accuracy greatly.

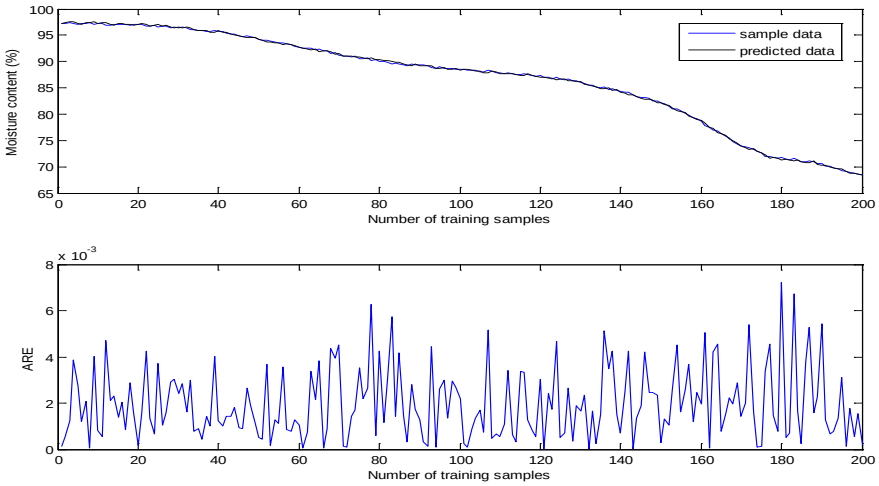

Figure 9: Predicted results of RELM model on training samples

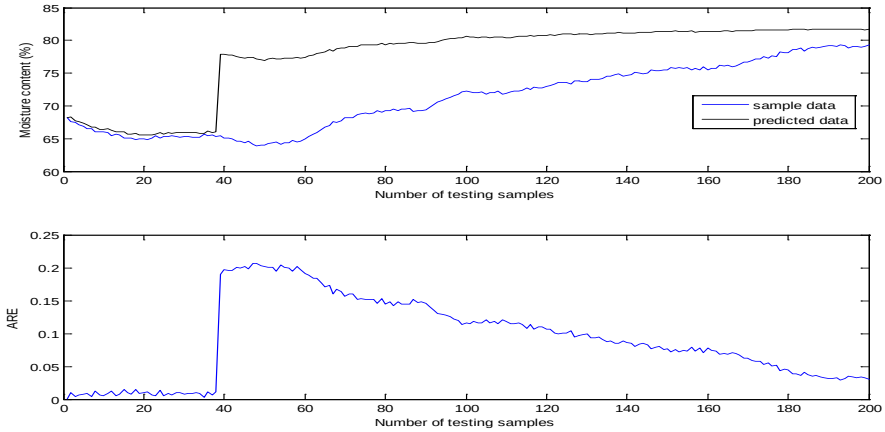

Figure 10: Predicted results of RELM model on testing samples without update

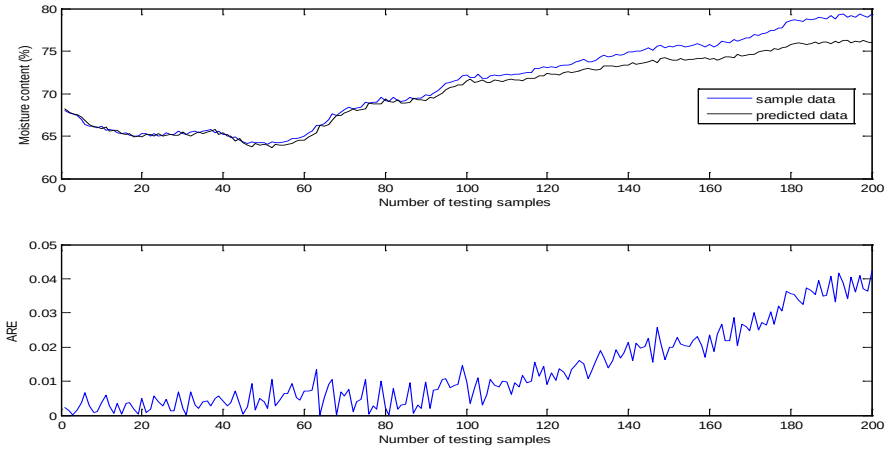

Figure 11: Predicted results of RELM model on testing results after the first update

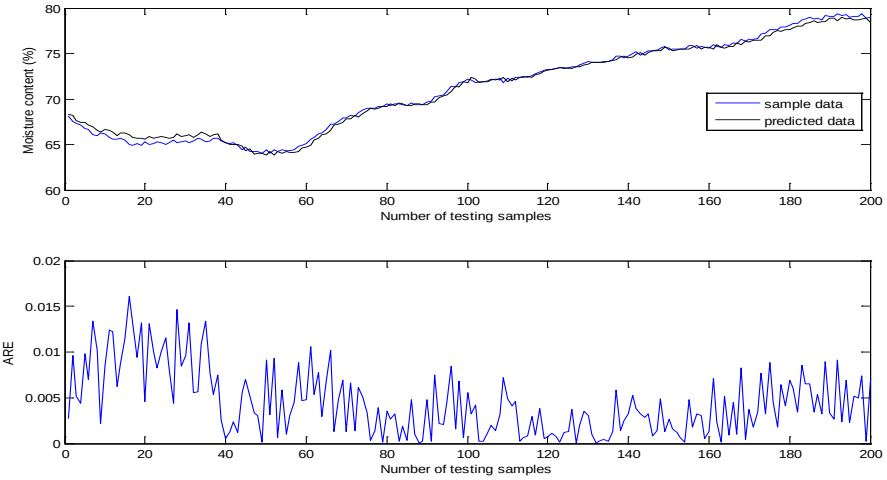

Figure 12: Predicted results of RELM model on testing samples after the second update

When the final GP and RELM models are determined, they can be regarded as components of IMM predictor for a hybrid prediction. Fig. 13 illustrates the predicted moisture content curves of different models. And the corresponding RMSE and RRMSE are shown in Tab. 3. It demonstrates that IMM hybrid prediction shows better prediction performance on accuracy and robustness. In fact, adding meaningful new samples can reduce the impact from sudden changes of operating conditions, but it cannot break the limits from the model itself. Comparing with single-model prediction, hybrid prediction can combine the characteristics of different models and avoid the error accumulation resulting from single-model prediction.

Table 2: RMSE and RRMSE of RELM model for prediction

RELM	Without update	After the first update	After the second update
RMSE	7.5965	1.7433	0.6950
RRMSE (%)	11.05	2.05	0.77

Table 3: RMSE and RRMSE of different models for prediction

Model	GP	RELM	IMM
RMSE	0.4126	0.6950	0.3771
RRMSE (%)	0.41	0.77	0.38

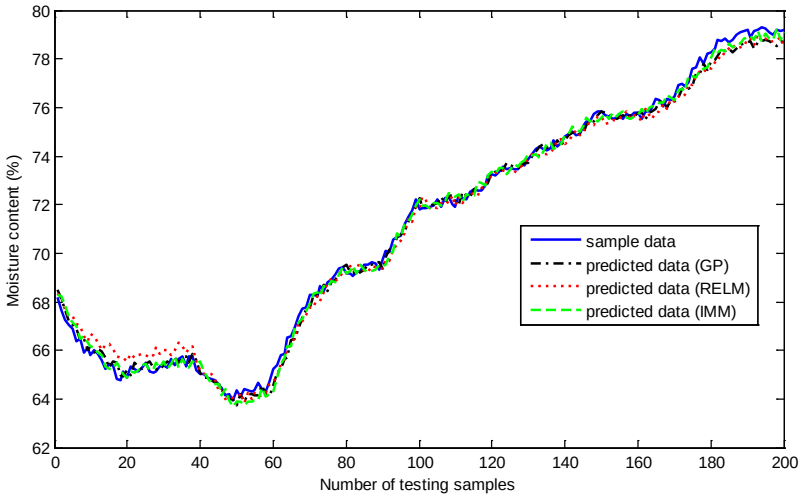

Figure 13: Predicted results of different models on testing samples

6 Conclusion

An interacting multiple-model prediction method is proposed and applied to moisture content of ASP flooding in this paper. Firstly, GP model and RELM model are used to model the process respectively on the basis of their own characteristics. The training results demonstrate that both models match the samples very well. However, the predictive accuracy may become bad when the operating conditions change. Thus, meaningful new sample data are incorporated into the models by recursive update. The testing results show the effectiveness of update strategies. Then, IMM algorithm is utilized for the hybrid prediction using the two models as components. The predicted results illustrate the hybrid prediction outperforms each single prediction on accuracy and robustness. The moisture content is a very important parameter in the process of oil development. It is no doubt that the proposed method can be viewed as an effective tool for moisture content prediction.

Acknowledgement: Authors would like to thank the associate editor and the anonymous referees for their valuable comments and suggestions. This work is supported by National Natural Science Foundation under Grant No. 60974039, National Natural Science Foundation under Grant No. 61573378, Natural Science Foundation of Shandong province under Grant No. ZR2011FM002, the Fundamental Research Funds for the Central Universities under Grant No. 15CX06064A.

Conflict of interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

Ažman, K.; Kocijan, J. (2007): Application of gaussian processes for black-box modelling of Biosystems. *ISA transactions*, vol. 46, no. 4, pp. 443-457.

Chan, L. L. T.; Liu, Y.; Chen, J. (2013): Nonlinear system identification with selective recursive gaussian process models. *Industrial & Engineering Chemistry Research*, vol. 52, no. 51, pp. 18276-18286.

Csató, L.; Opper, M. (2002): Sparse on-line gaussian processes. *Neural computation*, vol. 14, no. 3, pp. 641-668.

Deisenroth, M. P.; Rasmussen, C. E.; Peters, J. (2009): Gaussian process dynamic programming. *Neurocomputing*, vol. 72, no. 7, pp. 1508-1524.

Deng, W.; Zheng, Q.; Chen, L. (2009): Regularized extreme learning machine. *2009 IEEE Symposium on Computational Intelligence and Data Mining*, pp. 389-395.

di Sciascio, F.; Amicarelli, A. N. (2008): Biomass estimation in batch biotechnological processes by bayesian Gaussian process regression. *Computers & Chemical Engineering*, vol. 32, no. 12, pp. 3264-3273.

Golub, G. H.; Van Loan, C. F. (2012): *Matrix computations*, vol. 3, JHU Press.

Gregorčič, G.; Lightbody, G. (2008): Nonlinear system identification: From multiple-model networks to Gaussian processes. *Engineering Applications of Artificial Intelligence*, vol. 21, pp. 7, pp. 1035-1055.

Gregorčič, G.; Lightbody, G. (2009): Gaussian process approach for modelling of nonlinear systems. *Engineering Applications of Artificial Intelligence*, vol. 22, no. 4, pp. 522-533.

Huang, G. B.; Zhu, Q. Y.; Siew, C. K. (2006): Extreme learning machine: theory and applications. *Neurocomputing*, vol. 70, no. 1, pp. 489-501.

Hong, P. T.; Tran, V. T.; Yang, B. S. (2010): A hybrid of nonlinear autoregressive model with exogenous input and autoregressive moving average model for long-term machine state forecasting. *Expert Systems with Applications*, vol. 37, no. 4, pp. 3310-3317.

Keerthi, S.; Chu, W. (2005): A matching pursuit approach to sparse gaussian process regression. *Advances in neural information processing systems*, pp. 643-650.

Kocijan, J.; Likar, B. (2008): Gas-liquid separator modelling and simulation with gaussian-process models. *Simulation Modelling Practice and Theory*, vol. 16, no. 8, pp. 910-922.

Li, D.; Xie, Q.; Jin, Q. (2015): Quasi-linear extreme learning machine model based nonlinear system identification. *Proceedings of ELM-2014*, Springer, vol. 1, pp. 121-130.

Li, X. L.; Jia, C.; Liu, D. X.; Ding, D. W. (2014): Adaptive control of nonlinear discrete-time systems by using os-elm neural networks. *Abstract and Applied Analysis*, pp. 1-11.

Ni, W.; Tan, S. K.; Ng, W. J.; Brown, S. D. (2012): Moving-window gpr for nonlinear dynamic system modeling with dual updating and dual preprocessing. *Industrial & Engineering Chemistry Research*, vol. 51, no. 18, pp. 6416-6428.

O'Hagan, A.; Kingman, J. (1978): Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society*: *Series B (Methodological)*, pp. 1-42.

Qi, C.; Li, H. X. (2008): A karhunen-Loève decomposition-based wiener modeling approach for nonlinear distributed parameter processes. *Industrial & Engineering Chemistry Research*, vol. 47, no. 12, pp. 4184-4192.

Qi, C.; Li, H. X. (2009): A time/space separation-based hammerstein modeling approach for nonlinear distributed parameter processes. *Computers & Chemical Engineering*, vol. 33, no. 7, pp. 1247-1260.

Rasmussen, C. E. (1996): *Evaluation of Gaussian processes and other methods for non-linear regression (Ph.D. thesis)*. Citeseer.

Rasmussen, C. E.; Nickisch, H. (2010): Gaussian Processes for Machine Learning (GPML) Toolbox. *Journal of Machine Learning Research*, vol. 11, no. 6, pp. 3011-3015.

Seeger, M.; Williams, C.; Lawrence, N. (2003): Fast forward selection to speed up sparse gaussian process regression. *Artificial Intelligence and Statistics*.

Shao, Z.; Er, M. J. (2016): An online sequential learning algorithm for regularized extreme learning machine. *Neurocomputing*, vol. 173, pp. 778-788.

Shutang, G.; Qiang, G. (2010): Recent progress and evaluation of asp flooding for eor in daqing oil field. *SPE EOR Conference at Oil & Gas West Asia*, *Society of Petroleum Engineers*.

Tang, Y.; Li, Z.; Guan, X. (2014): Identification of nonlinear system using extreme learning machine based hammerstein model. *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 9, pp. 3171-3183.

Toh, K. A. (2008): Deterministic neural classification. *Neural computation*, vol. 20, no. 6, pp. 1565-1595.

Wang, M.; Qi, C.; Yan, H.; Shi, H. (2016): Hybrid neural network predictor for distributed parameter system based on nonlinear dimension reduction. *Neurocomputing*, vol. 171, pp. 1591-1597.

Wang, M.; Yan, X.; Shi, H. (2013): Spatiotemporal prediction for nonlinear parabolic distributed parameter system using an artificial neural network trained by group search optimization. *Neurocomputing*, vol. 113, pp. 234-240.

Wang, Y.; Wang, J.; Wei, X. (2015): A hybrid wind speed forecasting model based on phase space reconstruction theory and markov model: A case study of wind farms in northwest china. *Energy*, vol. 91, pp. 556-572.

Yan, W.; Shao, H.; Wang, X. (2004): Soft sensing modeling based on support vector machine and Bayesian model selection. *Computers & Chemical Engineering*, vol. 28, no. 8, pp. 1489-1498.

Zerpa, L. E.; Queipo, N. V.; Pintos, S.; Salager, J. L. (2005): An optimization methodology of alkaline-surfactant-polymer flooding processes using field scale numerical simulation and multiple surrogates. *Journal of Petroleum Science and Engineering*, vol. 47, no. 3, pp. 197-208.

Zhang, Y. (2011): Hourly traffic forecasts using interacting multiple model (imm) predictor. *IEEE Signal Processing Letters*, vol. 18, no. 10, pp. 607-610.